CL 00-0552
FINAL

Mixtrinsic Evolution

Adrian Stoica, Ricardo Zebulum and Didier Keymeulen

Center for Integrated Space Microsystems
Jet Propulsion Laboratory
California Institute of Technology
Pasadena CA 91109, USA

Abstract. Evolvable hardware (EHW) refers to automated synthesis/optimization of HW (e.g. electronic circuits) using evolutionary algorithms. *Extrinsic EHW* refers to evolution using software (SW) simulations of HW models, while *intrinsic EHW* refers to evolution with HW in the loop, evaluating directly the behavior/response of HW. For several reasons (including mismatches between models and physical HW, limitations of the simulator and testing system, etc.) circuits evolved in SW may not perform the same way when implemented in HW, and vice-versa. This *portability problem* limits the applicability of SW evolved solutions, and on the other hand, prevents the analysis (in SW) of solutions evolved in HW. This paper introduces a third approach to EHW called *mixtrinsic EHW (MEHW)*. In MEHW evolution takes place with hybrid populations in which some individuals are evaluated intrinsically and some extrinsically, within the same generation or in consecutive ones. A set of experiments using a Field Programmable Transistor Array (FPTA) architecture is presented to illustrate the portability problem, and to demonstrate the efficiency of mixtrinsic EHW in solving this problem.

1 Introduction

Evolvable HW (EHW) refers to automated synthesis/optimization of HW (e.g. electronic circuits) using evolutionary algorithms. Previous reports remark that solutions obtained by evolutionary design suffer from what it will be referred in this paper as *the portability problem*. For example, it was observed that some circuits obtained through evolutionary design on one HW platform had a different behavior when tested on a second platform, although the two were of similar type/construction [1]. Furthermore, a similar situation was encountered when attempting to port to HW the result of a solution evolved in SW, and vice-versa [2].

Evolution based on simulations of HW models is referred to as *extrinsic evolution (EE)*, while evolution with the HW in the loop (evaluating directly the behavior/response of HW) is referred to as *intrinsic evolution (IE)*. Successful accounts of intrinsic EHW (IEHW) and extrinsic EHW (EEHW) are reported in the literature [3-5]. However the portability between the SW and HW implementations of the solutions has been difficult. Researchers who evolved extrinsically often lacked suitable programmable devices to test their solutions (particularly for the evolution of analog circuits). In turn, those evolving intrinsically had limited access to the SW models of their HW platform, which often was proprietary information. Only

recently, and mainly through evolving on in-house built devices/test-boards, it became apparent that mismatches may exist, and the solutions evolved in SW may not hold in HW and vice-versa [2]. To solve this portability problem, a third approach, called *mixtrinsic EHW*, is proposed in this paper. Mixtrinsic EHW encompasses a family of techniques that combine the intrinsic and extrinsic modes in a variety of ways. The most straightforward alternative is the use of a mixed population of both SW models and reconfigurable HW.

This paper illustrates the portability problem with several examples, and presents a solution to this problem using mixtrinsic evolution (ME). The paper is organized as follows: Section 2 discusses characteristic aspects of extrinsic and intrinsic evolution. Section 3 focuses on the portability problem. Section 4 introduces a novel approach to EHW called mixtrinsic EHW (MEHW). Section 5 reviews the main characteristics of a Field Programmable Transistor Array (FPTA) used in the following sections as an evolutionary platform to demonstrate MEHW. Section 6 illustrates the portability problem and demonstrates how MEHW can solve it by exploiting common characteristics of the SW and HW. It also discusses the opposite effect, i.e. how MEHW can be used to emphasize differences between SW and HW and possibly capitalize on characteristics of physical HW. Section 7 presents the conclusions.

2 Extrinsic and Intrinsic Evolution

Two approaches to EHW have been proposed. The first uses *extrinsic* evolution, the candidate solutions are evaluated as SW models (of HW) and evaluations are done using a simulator. EEHW is schematically illustrated in Fig. 1. The population is homogeneous, and consists of SW models (e.g. SPICE netlists) that describe an electronic circuit to a certain degree of accuracy. The second approach to EHW uses intrinsic evolution, where the candidate solutions are in the form of physical HW configurations on programmable devices/architectures, which are evaluated using some test/evaluation equipment. IEHW is illustrated in Fig. 2. IEHW is more sensitive as the candidate solutions can a) damage the chip in some overstressing conditions, and b) be affected by previously configured/tested candidates. While in EEHW, individual candidates have no influence on each other, in IEHW they do, because each candidate performs on the same "stage" as its predecessors, which may have left an imprint on the "stage". For example, the charge accumulated during the evaluation of the performance of one individual affects the behavior of the next.

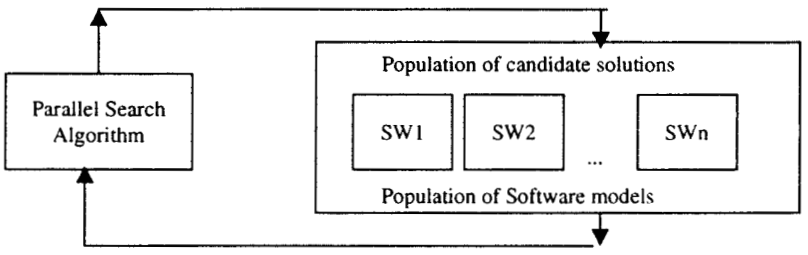

Fig. 1. Extrinsic EHW: evaluations of software solutions

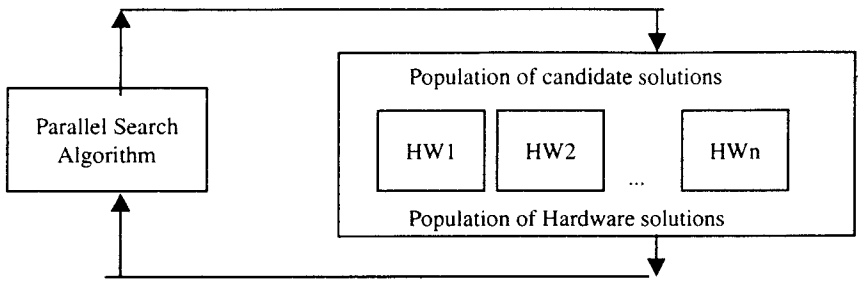

Fig. 2. Intrinsic EHW: evaluations of hardware solutions

3 The Portability Problem

Early experiments in EHW made it apparent that the solutions obtained by evolutionary design may suffer from a portability problem. For example, it was observed that some circuits obtained through evolutionary design on one HW platform had a different behavior when tested on a second platform, although the two were of similar type/construction. Thus, a circuit evolved on a corner of an FPGA did not reproduce the same behavior when it was implemented on a different part of the same FPGA [1]. Another situation is related to porting to HW a circuit evolved in SW (or vice-versa validating in SW a solution evolved in HW) as reported in [2]. Some of the circuits resulting as solutions from extrinsic evolution do not produce the same correct response when implemented/ported into HW. Vice-versa, many of the circuit topologies resulting from intrinsic evolution do not produce a good response (as obtained in the real HW) when they are simulated in SW.

One reason behind the portability problem is that, in each case, evolution finds the easy way out, optimizing for whichever raw material is given. The portability problem between two HW platforms is strongly related to differences in a set of characteristics that evolution exploited in one platform and can not exploit in the second. The difference between the response of a SW evaluation and HW evaluation of two circuits described by the same chromosome may be caused by one or more factors originated either in the phenotype itself or in the way this is observed/evaluated. In some experiments, in particular when floating gate based solutions were involved (unusual for human designs), circuits may appear good during evaluation in the rapid sequence of tests of individuals in the population. However when the individual is evaluated alone, statically, it may not perform as well. (The following discussion refers to Fig 3, illustrating the evaluation paths in EEHW and IEHW). Some of these factors are summarized in the following:

1. Differences between model and real HW: a) Simplified models (e.g. to gain speed in SPICE runs), b) Incomplete models because of lack of information about fabrication, c) HW can change from the moment it was modeled/identified (temperature, radiation, operating conditions), d) HW can change in time after evaluation (e.g. slow discharge)
2. Simulator limitations (SW evaluation): a) Convergence conditions, which humans may be able to help by setting/adjusting values, b) Conditions

unknown *a-priori* (e.g. charges, initial conditions), in which case the system of differential equations can not be solved

3. HW testing limitations: a)Transients, b) Charge, e.g. remaining from a previously evaluated individual, c) Impedance loading of measured circuit, d) Time delays between physical signals (e.g. excitation) and outputs, e) Artifacts originating in signal generators, data acquisition paths, sampling, A/D, etc.

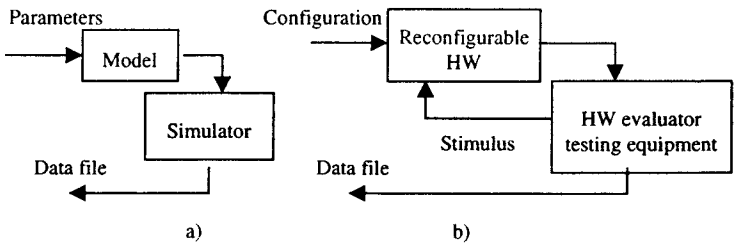

Fig. 3. Path from chromosome to behavior data file a)extrinsic and b)intrinsic

The problems of portability to HW raises questions about the usefulness of the extrinsic solution from a practical/pragmatic/commercial point of view. On the other hand, not being able to simulate the solution evolved in HW largely diminishes the confidence in the intrinsic solution as it can not be analyzed and can not be proven or is not guaranteed to work outside the operating region used in the evaluations during evolution. When a solution exploits very specific effects, there may be situations where its applicability range may be very limited. This type of solution is a "point design", while what is needed is a "domain-wide" design, able to characterize a solution within a large envelope along several parameters: temperature, power supply, radiation effects, etc. In an attempt to achieve this type of robustness, it was proposed to evaluate each solution over a complete domain [4]. In our opinion the practicality of this approach is limited to simple cases due to cost issues, especially for space/military qualification. To deal with portability between two HW platforms, Thompson proposed to use different testing HW for different individuals in the evolutionary run. This can be done e.g. changing the area on the chip where the circuit is evaluated or moving between different chips, to ensure that only solutions that perform well on all platforms are selected during evolution. There is no guaranty that a newly evolved solution on one chip will work on others. One can not analyze/validate it in SW either if no accurate SW model is available (including parasitic effects that may be exploited by evolution as in [4]).

4 Mixtrinsic Evolution

Mixtrinsic evolution (ME) relates to evolving on mixed/heterogeneous populations, composed partly of models and partly of real HW [6]. This would constrain evolution to a solution that jointly simulates well in SW, and performs well in HW, i.e. a solution that exploits only the HW characteristics included in the SW model for producing the desired behavior (see Fig. 4). Solutions based on HW properties outside of the SW model are eliminated by evolution. In ME the population

of candidate solutions is robust, more likely to be in agreement with common design rules, and, if novel, more likely to be patentable (i.e. to have generality and not depend on a fabrication process). The greatest advantage of the resulting solution is that it can both operate in HW and be analyzed in SW to explore its behavior outside the domain within which it was evolved. This is the only way to have insights and confidence in the evolved HW solution. Also, the resulting circuit is more likely to be portable to other HW platforms.

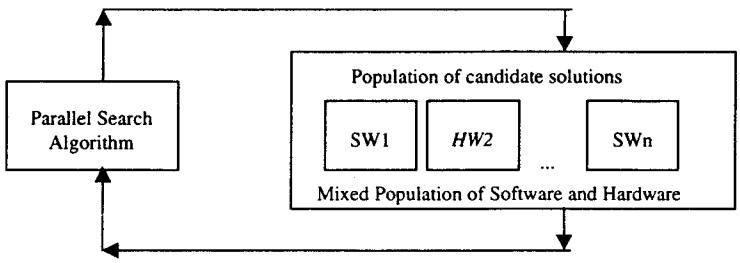

Fig. 4. Mixtrinsic EHW: evaluations of mixed populations comprised of both hardware and software solutions

Two types of ME are further detailed: *complementary and combined ME.* In *complementary* MEHW, candidate solutions are evaluated after being alternatively reassigned to either a HW or a SW platform (subject to random or deterministic choice). For example, an individual in a generation would have probability P to be evaluated in HW and probability (1-P) to be evaluated in SW. Assuming HW evaluates faster than SW one can speed-up evaluations by having a high value of P, which will cause a larger population to be evaluated in HW. The probability P, and related to it the ratio of individuals evaluated in HW over the total population, could also be variable parameters, adjustable during evolution.

In what we refer here as *combined* MEHW, each individual is evaluated both in HW and SW, and a combined fitness function is calculated. In the simplest case this can be a simple average of the two components or may involve adjustable weights etc.

We refer to the above description as a *matching* ME, for which the emphasis was on reinforcing the matching of similar characteristics of the SW models and the HW it describes. An opposite idea would be to reinforce dissimilarities and reinforce HW (or SW) distinctive characteristics, i.e. mismatches. We will refer to this as *mismatching* MEHW. This paper will demonstrate MEHW using a Field Programmable Transistor Array (FPTA) as evolutionary testbed.

5 FPTA Architecture

The FPTA was proposed as a flexible, versatile platform for EHW experiments and developed as an intermediate step toward a stand alone evolvable System-On-a-Chip (SOC) [7]. The architecture is cellular, and has similarities with other cellular architectures as encountered in FPGA (e.g. Xilinx 6200 family) or cellular neural

networks chips. The main distinguishing characteristic is related to the particular definition of the elementary cell. This paper uses the first version of the FPTA cell, as illustrated in Fig. 5 (a new version is currently under development.) The structure is largely a "sea of transistors" where transistors are interconnected by other transistors that act as signal passing devices (gray-level switches). Details of the FPTA, its HW implementation and evolutionary experiments on FPTA can be found in [7]. The flexibility and versatility of the FPTA in implementing a variety of building blocks used in electronic circuits, as well as a discussion in the context of other programmable devices can be found in [8].

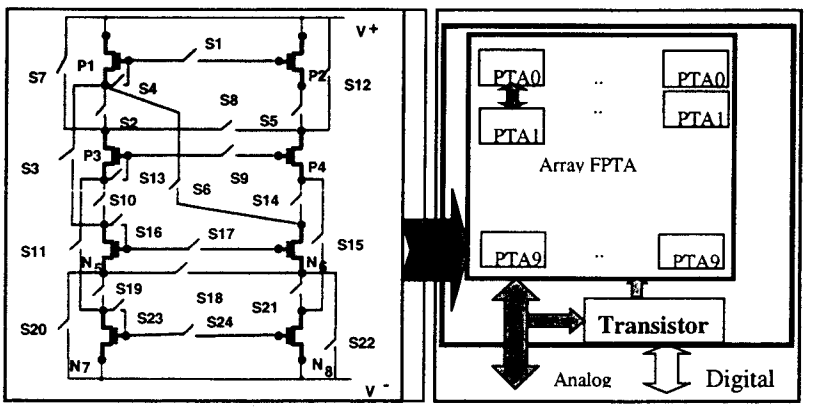

Fig.5 FPTA cell and FPTA chip

What is essential is that reconfiguration at transistor level allows definition of building blocks or subcircuits at a variety of levels of granularity. At lowest level one can configure subcircuits such as current mirrors and differential pairs, while more complex blocks such as logical gates, Operational Amplifiers (OpAmps), can also be easily configured. The level of granularity can be set by the designer, who can freeze the architecture to define high level components for evolution. Alternatively one can expect evolution to come with the building blocks that are the most suitable for the particular application (in the same sense as Koza's Automatic Defined Functions [4]).

The FPTA was exercised on a testbed that supports HW and SW evaluations (intrinsic/extrinsic). The SW subsystem makes use of the Caltech 256-processor HP Exemplar parallel computer to run multiple copies of SPICE. The HW subsystem is built around National Instruments LabView, associated data acquisition boards, signal generators, and other equipment, see [7] for more details.

6 Mixtrinsic evolution experiments: on convergent and divergent ME

This section exemplifies the portability problem and demonstrates the ME's ability to solve this problem. The following experiments are shown: a) Extrinsic evolution, with the resulting solution valid in SW but invalid when tested in HW, b) Intrinsic

evolution, with the resulting solution valid in HW but invalid when tested in SW, c) Mixtrinsic evolution, with the resulting solution valid both in SW and HW.

The experiments show the evolutionary synthesis of an AND gate, using one FPTA cell. The input signals and an acceptable output response are shown in Fig. 6. The level 'high' input signals corresponding to logical '1' were controlled to keep their level for 5 ms, which corresponds to 20 samples on LabView graphs of acquired signal from HW. All experiments (about 20 runs for each case in a) and b) and 5 each for c1) and c2) below) used 30 individuals for 30 generations.

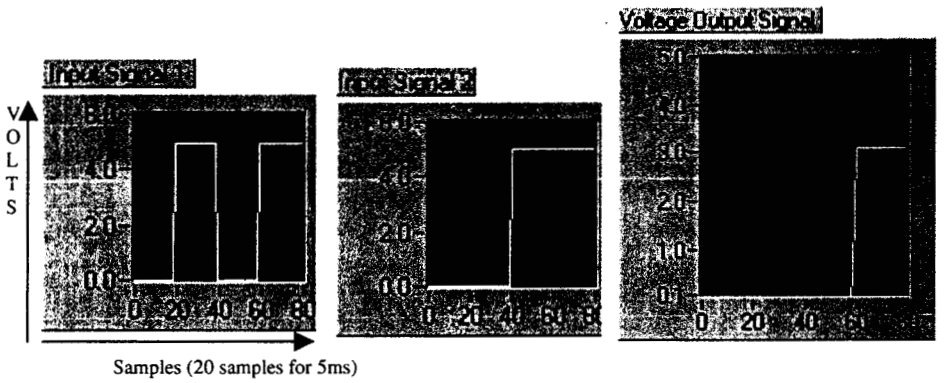

Fig. 6. Inputs and output for an AND gate

a) Extrinsic evolution. Two best individuals in the last generation are the solutions presented here for discussion. One of the solutions could be validated in HW. However, the circuit shown in Fig. 7, which is in fact the solution with the highest fitness, does not give satisfactory response when downloaded into HW. Thus, two direct observations can be made: a) one solution is validated in HW while the other is not, b) in this particular case, the "better " (in the sense of the fitness function that rewarded for higher value of the '1' level) solution in SW performs worse in HW. It appears that the solutions obtained through extrinsic evolution may not work in HW. Moreover, in many cases, there is no way to know for sure if it works without validating in actual HW.* (* We believe this reflects the current state-of-the-art, but admittedly we are strongly biased by our own experience with a certain model and HW. We believe that increasingly higher confidence in a solution would come from minimizing the negative effects of the factors discussed in Section 3. We also refer mainly to effects in analog circuits, and especially to those NOT relying on well understood building blocks, such as Op. Amps etc).

b) Intrinsic evolution. A circuit obtained intrinsically (best individual after a run with 30 individuals and 30 generations) and its response in HW and SW are shown in Fig. 8. The conclusion is that the solutions obtained through intrinsic evolution may not work in SW (see circuit responses in SW and in HW in the figure).

c1) Combined Mixtrinsic Evolution (Matching). Each individual was evaluated both in HW and SW. The combined fitness was a simple average. The SW and HW responses are similar. The resulting solution is shown in Fig. 9.

c2) Complementary Mixtrinsic Evolution (Matching). Each individual was allocated either to HW or SW evaluation with a 50% probability. The response of the resulting solution is identical to that illustrated in Fig. 9 (although the circuit is slightly different) and is omitted for space reasons.

In all experiments, the best 6 individuals of the last generation were tested both in HW and SW and they displayed similar responses. Although this is only empirical evidence, there is a good reason to believe that selection pressure would indeed favor solutions that display similar response in HW and SW.

d) Divergent ME. exploiting the distinctive characteristics of HW (or SW): Once accounted for the likelihood of obtaining mismatched responses between HW and SW, it appears straightforward to accept that selection pressure can force things in this direction (of mismatches). We are currently performing experiments in which we use combined evolution (each individual is evaluated twice, once in HW and once in SW). The combined fitness functions are either ratios of fitness of HW over fitness of SW, or its derivations as the sum of HW fitness and the inverse of SW fitness. Preliminary experiments illustrate that, indeed, resulting circuits produce the expected result in HW, while not being able to give a good response in SW.

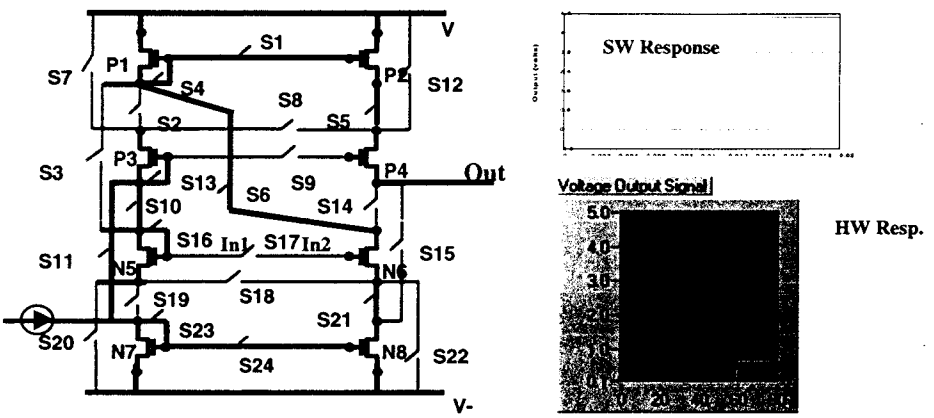

Fig. 7. Extrinsically evolved circuit, its response in SW and invalid response in HW

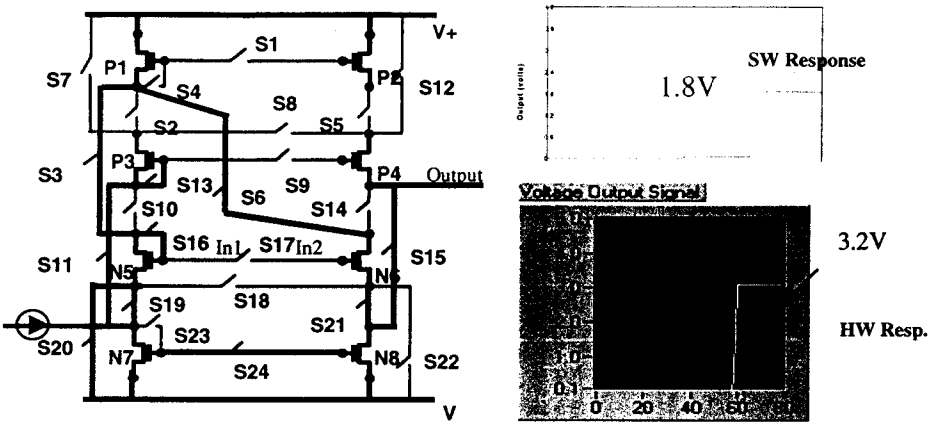

Fig. 8. Intrinsically evolved circuit, its response in HW and its invalid response in SW

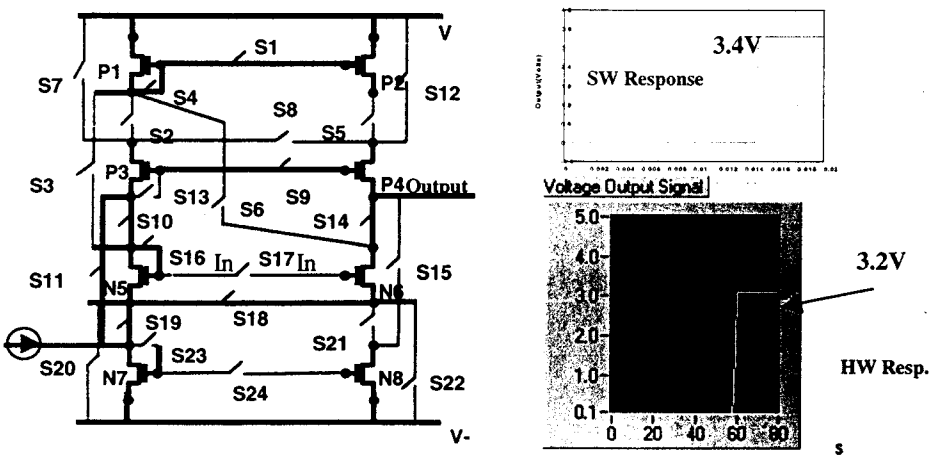

Fig. 9. Circuit obtained by mixtrinsic evolution, its valid responses in SW and in HW

In the graphs of Figures 7, 8 and 9, the axis represent samples (X) and response in volts (Y).

7 Conclusion

Both Intrinsic and Extrinsic EHW appears to suffer from a portability problem (solutions evolved in SW do not run in HW and vice-versa), which is here illustrated through evolutionary experiments for the synthesis of an AND gate. A new approach introduced here and referred to as *mixtrinsic* evolution uses heterogeneous populations of individuals, some of which are evaluated extrinsically and some intrinsically. *Convergent mixtrinsic* evolution reinforces similarities between SW and HW behavior. Two flavors of the convergent style are demonstrated: complementary (population is mixed within the same generation, each individual being randomly evaluated either in HW or SW) and *combined* (each individual is evaluated both in HW and SW and a combined fitness is assigned). The demonstration uses Field Programmable Transistor Array architecture and shows that all the best individuals evolved in this way are validated both in HW and in SW. The opposite flavor of *mixtrinsic* evolution introduced here is *divergent* evolution, in which case selection rewards the distinctions between HW and SW, e.g. forcing circuits that exploit HW characteristics not modeled in SW.

Acknowledgements

The research described in this paper was carried out by the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics Space Administration.

References

1. Thompson, A. Silicon Evolution. In: Proceedings of Genetic Programming 1996 (GP96), J.R. Koza et al. (Eds), pages 444-452, MIT Press 1996
2. Stoica, A., Keymeulen, D., Tawel, R., Salazar-Lazaro, C. and Li, W. (1999) "Evolutionary experiments with a fine-grained reconfigurable architecture for analog and digital CMOS circuits. *Proceedings of the First NASA/DOD Workshop on Evolvable Hardware*, Pasadena, CA, July 19-21, IEEE Computer Society Press, pp. 76-84
3. Koza, J., F.H. Bennett, D. Andre, and M.A Keane, "Genetic Programming: Darwinian invention and problem solving", Morgan Kaufmann Publishers, San Francisco, CA, 1999
4. Thompson, A., Layzell, P. and Zebulum, R., Explorations in design space: unconventional electronics design through artificial evolution, IEEE Transactions on Evolutionary Computation, September 1999, V.3, N.3 pp. 167-196
5. Higuchi, T. et al., Real-world applications of analog and digital evolvable hardware, IEEE Transactions on Evolutionary Computation, September 1999, V.3, N.3 pp. 220-235
6. Stoica, A. Evolutionary design technique using a mixed population of candidate solutions, some of which are evaluated in SW and some in reconfigurable HW, New Technology Report NPO 20733, JPL, California Institute of Technology, July 1999
7. Stoica, A. Towards Evolvable Hardware Chips: Experiments with a Programmable Transistor Array. Proc.7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired SYstems, *Microneuro'99*,. Granada, Spain, April 7-9, 1999, pp. 156-162
8. Zebulum, R. Stoica, A. and Keymeulen, D., A flexible model of CMOS Field Programmable Transistor Array targeted to Evolvable Hardware, to appear in the 3rd Int. Conf. on Evolvable Systems, ICES2000, Edinburgh, Scotland, April, 2000.